

ESTRUCTURAS DE DATOS Y ALGORITMOS



COLECCIÓN TEXTOS DOCENTES

FICHA CATALOGRÁFICA

CAMPOS LACLAUSTRA, Javier
Estructuras de datos y algoritmos / Javier Campos Laclaustra. — Zaragoza :
Prensas Universitarias de Zaragoza, 1995
270 p. ; 24 cm. — (Textos Docentes ; 41)
Bibliografía

ISBN 84-7733-447-1

1. Programación de ordenadores—Tratados, manuales, etc. 2. Lenguajes de
ordenadores—Tratados, manuales, etc. 3. Algoritmos—Tratados, manuales, etc. I.
Prensas Universitarias de Zaragoza, ed. II. Título III. Serie: Textos Docentes
(Prensas Universitarias de Zaragoza) ; 41
681.31.06:510.5

No está permitida la reproducción total o parcial de este libro, ni su tratamiento informático, ni la transmisión de ninguna forma o por cualquier medio, ya sea electrónico, mecánico, por fotocopia, por registro u otros métodos, ni su préstamo, alquiler o cualquier forma de cesión de uso del ejemplar, sin el permiso previo y por escrito de los titulares del Copyright.

- © Javier Campos Laclaustra
- © De la presente edición, Prensas Universitarias de Zaragoza
1.ª edición, 1995

Editado por Prensas Universitarias de Zaragoza
Edificio de Ciencias Geológicas
C/ Pedro Cerbuna, 12
50009 Zaragoza, España

Prensas Universitarias de Zaragoza es el sello editorial de la Universidad de Zaragoza, que edita e imprime libros desde su fundación en 1583.

Impreso en España
Imprime: Servicio de Publicaciones. Universidad de Zaragoza
D.L.: Z-3233-95

ESTRUCTURAS DE DATOS Y ALGORITMOS

Javier Campos Laclaustra



Pr n s s Un v r s t r s Z r z

ESTRUCTURAS DE DATOS Y ALGORITMOS

Prólogo

La *abstracción de acciones* es la base de la metodología de *diseño descendente por refinamientos sucesivos*, útil para la resolución de pequeños problemas de tratamiento de información. Sin embargo, para afrontar la construcción de programas en media y gran escala es necesaria una metodología de *diseño modular*, que permita la partición del trabajo en unidades de programa que puedan ser desarrolladas independientemente del resto. El propósito de estos apuntes es presentar los principios básicos de una metodología de diseño modular basada en la *abstracción de datos*.

Este material ha sido elaborado para servir como soporte de la asignatura *Estructuras de datos y algoritmos*, que se imparte en el tercer semestre de los estudios de Ingeniería Informática en el Centro Politécnico Superior de la Universidad de Zaragoza. Los alumnos que cursan dicha asignatura han seguido previamente dos semestres de programación en los que han debido aprender a especificar formalmente y a diseñar programas en pequeña escala, utilizando tipos de datos sencillos (como los predefinidos en un lenguaje de programación de la familia del *Pascal*); los alumnos conocen técnicas de diseño recursivo e iterativo, así como las herramientas básicas para poder medir la eficiencia de los algoritmos (atendiendo a su tiempo de ejecución). No obstante, el material presentado puede ser útil también para un segundo nivel en todos aquellos planes de estudios en los que se incluyan dos cursos de programación de computadores.

Pueden encontrarse en las librerías varios trabajos (muchos de ellos ya clásicos) con títulos similares o iguales a éste. Sin embargo, y ésta es la razón para la existencia de uno nuevo, la aproximación al tema que se pretende desarrollar es bien diferente. De hecho, el título que el autor habría elegido, en caso de no haber optado por mantener el nombre de la asignatura antes mencionada, hubiese sido más bien *Tipos abstractos de datos y algoritmos* o mejor *Introducción a la programación con tipos abstractos de datos*. La diferencia estriba en el énfasis que se pretende dar en las páginas que siguen a la especificación formal de los tipos (de ahí la expresión «tipos abstractos de datos») como herramienta fundamental para el diseño modular de programas, en lugar de limitarse a presentar las estructuras de datos necesarias para representar los valores de los tipos definidos.

El comentario anterior no debe hacer pensar al lector que el material que sigue es original del autor. Nada más lejos de la realidad. Únicamente nos hemos limitado a enlazar las excelentes aproximaciones existentes en la literatura a la definición y conceptos relacionados con los tipos abstractos de datos y su especi-

ficación algebraica (véanse, por ejemplo, los dos últimos capítulos de la obra de Ricardo Peña titulada *Diseño de Programas. Formalismo y Abstracción*) con los trabajos más clásicos sobre estructuras de datos y algoritmos de manipulación (como, por ejemplo, *Estructuras de Datos y Algoritmos*, de Aho, Hopcroft y Ullman).

Los apuntes están estructurados en lecciones, agrupadas en grandes temas. En el primero de ellos, titulado «Tipos abstractos de datos», se presentan los conceptos fundamentales sobre los tipos abstractos de datos, su especificación formal (algebraica) y su utilización en el diseño modular de programas.

El segundo tema, «Tipos de datos lineales», introduce tres de los tipos abstractos lineales más representativos y útiles en programación: las pilas, las colas y las listas con acceso por posición. Para cada nuevo tipo presentado se incluyen su especificación formal, una o varias soluciones para la representación de sus valores, la implementación de las operaciones más importantes, su coste computacional y algunos ejemplos de aplicación.

El tercer tema, titulado «Árboles y esquemas algorítmicos», incluye los detalles sobre algunos de los tipos de árboles más frecuentemente utilizados, como los árboles binarios, árboles ordenados, árboles de búsqueda, montículos..., y ejemplos de aplicación. Además, se introducen los algoritmos de vuelta atrás y las heurísticas voraces.

Los dos últimos temas, sobre «Tipos de datos funcionales» (o *tablas*) e «Introducción a los grafos», no se desarrollan con la misma extensión que los anteriores por razones diferentes. En el caso de las tablas, tras las definiciones formales convenientes, se hace hincapié en la representación mediante tablas dispersas basadas en la utilización de una función de localización (*hashing*, en inglés) y en las tablas multidimensionales representadas mediante estructuras de listas múltiples, pues otras representaciones posibles basadas en listas lineales o árboles de búsqueda no precisan mayor explicación tras el estudio de los temas previos.

En cuanto a los grafos, los alumnos de Ingeniería Informática (a quienes va dirigida preferentemente esta obra) han cursado previamente una asignatura titulada *Matemática discreta*, en la que se les ha presentado el concepto de grafo y una buena colección de algoritmos para su manipulación. Por ello, y atendiendo a razones de completitud, se presentan sólo las especificaciones formales y varias alternativas de representación, junto a algunas consideraciones sobre el efecto que la elección de la representación tiene en el coste de los algoritmos de manipulación.

Por último, un comentario sobre las notaciones empleadas y los lenguajes de programación que pueden servir como soporte de prácticas. Para la especificación

algebraica de tipos abstractos, se utiliza una sintaxis similar a la del lenguaje *OBJ*, pero en español. En cuanto a los módulos, estructuras de datos y algoritmos, se emplea una notación algorítmica, también en español, que consiste en una extensión modular de la notación utilizada en los apuntes sobre *Introducción a la programación*, elaborados por Javier Martínez y Javier Campos como soporte a la asignatura de igual nombre existente en el currículum de Ingeniería Informática del CPS.

En cuanto al lenguaje de programación soporte de las prácticas, el autor desaconseja la utilización de las extensiones modulares de Pascal (incluido el *Modula 2*), pues carecen de la posibilidad de definir tipos opacos y tipos genéricos, siendo ambos mecanismos fundamentales en la metodología desarrollada. Así, un lenguaje apropiado resulta ser el *Ada*, dotado de la posibilidad de definición de tipos opacos y tipos genéricos, con una sintaxis y una semántica bien pensadas y una dificultad de aprendizaje similar al Pascal, si se limita su presentación a la parte secuencial. Otras alternativas pueden encontrarse en lenguajes de programación *orientados a objetos* (como, por ejemplo, *C++*), dada la cercanía de los conceptos de *clase* y *tipo abstracto de dato*.

Javier Campos Laclaustra

Zaragoza, 30 de marzo de 1995

Bibliografía

- AHO, Alfred V.; John E. HOPCROFT; Jeffrey D. ULLMAN (1988): *Estructuras de Datos y Algoritmos*. Addison-Wesley Iberoamericana.
- BALCÁZAR, José Luis (1993): *Programación Metódica*. McGraw Hill.
- BARNES, John G.P. (1987): *Programación en Ada*. Ediciones Díaz de Santos.
- BRASSARD, Gilles y Paul BRATLEY (1990): *Algorítmica. Concepción y Análisis*. Masson.
- HOROWITZ, Ellis y Sartaj SAHNI (1984): *Fundamentals of Data Structures in Pascal*. Computer Science Press.
- KNUTH, Donald E. (1986): *El Arte de Programar Ordenadores. Volumen III: Clasificación y Búsqueda*. Reverté.
- McCRACKEN, Daniel D. (1987): *A Second Course in Computer Science with Pascal*. John Wiley & Sons.
- PEÑA MARÍ, Ricardo (1993): *Diseño de Programas. Formalismo y Abstracción*. Prentice Hall.
- WATT, David A. (1991): *Programming Language Syntax and Semantics*. Prentice Hall.
- WIRTH, Niklaus (1980): *Algoritmos + Estructuras de Datos = Programas*. Ediciones del Castillo.

Índice

Prólogo	7
TEMA I: TIPOS ABSTRACTOS DE DATOS	
Lección 1. Concepto, terminología y ejemplos	13
1.1. Concepto de abstracción	13
1.2. Tipos abstractos de datos	14
1.3. Ejemplo	16
Lección 2. Programación con TAD's.....	17
2.1. Los TAD's como base del diseño modular	17
2.2. Lenguajes de programación modular	20
2.3. La programación en gran escala.....	25
2.4. TAD's genéricos y algoritmos genéricos	28
Lección 3. Especificación algebraica de TAD's	33
3.1. Introducción	33
3.2. Signatura de una especificación algebraica	33
3.3. Ecuaciones de una especificación algebraica.....	35
Lección 4. Semántica de una especificación algebraica.....	39
4.1. Signatura y <i>SIG</i> -álgebras	39
4.2. Especificación y álgebra definida por ella	42
4.3. Modelos de una especificación	44
Lección 5. Construcción de especificaciones	49
5.1. Introducción	49
5.2. Operaciones: clasificación	49
5.3. Escritura de ecuaciones.....	51
5.4. Situaciones de error.....	53
5.5. Ejercicios.....	55
Lección 6. Verificación con especificaciones algebraicas.....	57
6.1. Introducción	57
6.2. Verificación de programas usuarios de TAD's.....	58
6.3. Corrección de implementaciones	62
TEMA II: TIPOS DE DATOS LINEALES	
Lección 7. El TAD pila. Implementación estática.....	67
7.1. Concepto de pila y especificación formal	67

7.2. Representación estática e implementación de operaciones.....	68
7.3. Representación de varias pilas en un vector	70
Lección 8. Datos puntero e implementación dinámica de pilas	73
8.1. Datos puntero y datos dinámicos	73
8.2. Estructuras de datos recursivas: representación mediante punteros y datos dinámicos	75
8.3. Representación dinámica de una pila e implementación de operaciones.....	77
Lección 9. Ejemplos de aplicación del TAD pila	81
9.1. Evaluación de expresiones postfijas	81
9.2. Traducción de expresiones infijas a postfijas.....	83
9.3. Recorrido de un laberinto.....	86
Lección 10. El TAD cola. Definición e implementación.....	91
10.1. Concepto de cola y especificación formal	91
10.2. Representación dinámica e implementación de operaciones.....	92
10.3. Representación estática circular.....	95
Lección 11. Ejemplos de aplicación del TAD cola.....	99
11.1. El problema de los palíndromos.....	99
11.2. Simulación de una cola de espera	100
Lección 12. Listas con acceso por posición: definición	107
12.1. Idea intuitiva y conjunto libre de generadoras	107
12.2. Especificación formal: caso de acceso por los extremos	108
12.3. Especificación formal: enriquecimiento con todas las operaciones	109
Lección 13. Listas con acceso por posición: implementación.....	111
13.1. Representación dinámica	111
13.2. Implementación de las operaciones	112
13.3. Mejoras en la representación.....	118
Lección 14. Caso de estudio: cálculo disperso	121
14.1. Mejoras en la representación de listas e implementación de operaciones.....	121
14.2. Vectores cuasi-vacíos	124
14.3. Matrices cuasi-vacías	130

TEMA III:

ÁRBOLES Y ESQUEMAS ALGORÍTMICOS

Lección 15. Árboles: concepto y especificación algebraica.....	139
15.1. Conceptos, definiciones y terminología básica.....	139
15.2. Especificación de árboles ordenados	140

15.3. Especificación de árboles binarios	143
Lección 16. Implementación de árboles binarios	145
16.1. Implementación estática.....	145
16.2. Implementación dinámica	146
Lección 17. Implementación de árboles ordenados.....	153
17.1. Implementación estática.....	153
17.2. Implementación dinámica	155
Lección 18. Árboles de búsqueda.....	163
18.1. Árboles binarios de búsqueda: especificación algebraica.....	163
18.2. Implementación de operaciones con árboles binarios de búsqueda.....	165
18.3. Árboles equilibrados, árboles m -arios de búsqueda y árboles B..	167
Lección 19. Colas con prioridades: representación con montículos y una aplicación	171
19.1. Cola con prioridades: concepto y especificación algebraica	171
19.2. Representación con montículos y coste de las operaciones	172
19.3. Representación estática de montículos e implementación de operaciones.....	175
19.4. Ejemplo de aplicación: método de ordenación basado en un montículo.....	179
Lección 20. Transformación de algoritmos recursivos en iterativos	183
20.1. Transformación de algoritmos recursivos finales	183
20.2. Transformación de algoritmos recursivos lineales (no finales) ..	185
20.3. Transformación de algoritmos recursivos múltiples: un caso particular	186
20.4. Transformación de algoritmos recursivos múltiples: caso general.....	190
Lección 21. Algoritmos de vuelta atrás y árboles de juego.....	197
21.1. Introducción al esquema de vuelta atrás	197
21.2. Ejemplo: el problema de las ocho reinas	199
21.3. Árboles de juego: planteamiento y ejemplo.....	202
21.4. Árboles de juego: estrategia <i>minimax</i>	204
Lección 22. Introducción a los algoritmos voraces	209
22.1. Estrategias voraces.....	209
22.2. Aplicación al problema del recorrido del caballo de ajedrez.....	210
22.3. Aplicación al problema del viajante.....	218

TEMA IV:
TIPOS DE DATOS FUNCIONALES

Lección 23. El TAD tabla: especificación y algunas implementaciones.....	223
23.1. Concepto de tabla.....	223
23.2. Especificación algebraica del TAD tabla	224
23.3. Algunas implementaciones sencillas	225
Lección 24. Tablas dispersas	227
24.1. Conceptos básicos	227
24.2. Funciones de localización	228
24.3. Resolución de colisiones por encadenamiento.....	230
24.4. Recolocación en el mismo vector soporte.....	234
Lección 25. Tablas multidimensionales.....	239
25.1. Concepto y ejemplo introductorio.....	239
25.2. Especificación algebraica de tablas bidimensionales.....	240
25.3. Implementación con estructuras de listas múltiples.....	242

TEMA V:
INTRODUCCIÓN A LOS GRAFOS

Lección 26. El TAD grafo	249
26.1. Conceptos básicos	249
26.2. Especificación algebraica.....	250
Lección 27. Implementaciones básicas de grafos.....	253
27.1. Matriz de adyacencia	253
27.2. Listas de adyacencia.....	254
27.3. Listas múltiples de adyacencia.....	255
Ejercicios propuestos	259
Bibliografía.....	265

